



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06343065 A**

(43) Date of publication of application: 13.12.94

(51) Int. Cl. **H04J 3/22**
G11B 20/12
H04B 14/04
H04N 5/76
H04N 5/93

(21) Application number: 06027634

(22) Date of filing: 25.02.94

(30) Priority: 26.02.93 JP 05 63293
16.03.93 US 93 32341

(71) Applicant: **SONY CORP**

(72) Inventor: **MAAKU FUERUTOMAN**

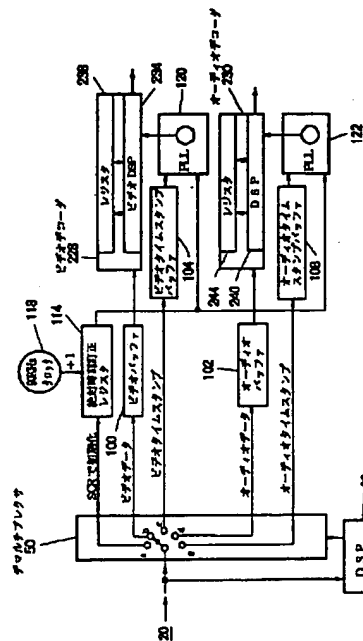
**(54) DECODING SYSTEM, DEMULTIPLEX METHOD
AND SIGNAL PROCESSING METHOD**

(57) Abstract:

PURPOSE: To easily decide a specific access unit to which a specific time stamp belongs.

CONSTITUTION: A local access unit counter used for digital AV application is included in a system which decodes the compressed video and audio signals. In this system, such access unit headers as a frame header, for example, an audio unit header or other data unit headers, etc., are lost or generated due to the storage medium errors and the transmission errors. The value so-called 'access unit count' is included in each elementary stream in order to suppress occurrence of the AV synchronization errors. This value is stored in the registers 238 and 244 respectively. The encoders 228 and 230 detect the loss and the errors of those access unit headers and can correct the AV synchronization errors. Then a specific bit can be added to show the discontinuity in the 'access unit count' and therefore the editing jobs can be supported.

COPYRIGHT: (C)1994,JPO



(11)特許出願公開番号

(43)公開日 平成6年(1994)12月13日

審査請求 未請求 請求項の数32 OL (全 19 頁)

(74)代理人 弁理士 稻本 義雄

【特許請求の範囲】

【請求項1】 タイムリファレンスを生成するタイムリファレンス生成手段と、

オーディオタイムスタンプがオーディオサンプル群と等しいか、または少ない数であり、かつ、ビデオタイムスタンプがビデオフレームと等しいか、または少ない数である場合において、コンボジットデータストリームを前記オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する分離手段と、

前記オーディオタイムスタンプとタイムリファレンスとを受け取り、前記タイムリファレンスを前記オーディオタイムスタンプと比較し、前記オーディオサンプル群の1つを前記オーディオタイムスタンプのそれぞれに関連付けるオーディオタイミング信号を生成するオーディオタイミング信号生成手段と、

前記オーディオサンプル群とオーディオタイミング信号を受け取り、前記オーディオサンプル群を前記オーディオタイミング信号に同期してデコードするオーディオデコード手段と、

前記ビデオタイムスタンプとタイムリファレンスとを受け取り、前記タイムリファレンスを前記ビデオタイムスタンプと比較し、前記ビデオフレームの1つを前記ビデオタイムスタンプの各々に関連付けるビデオタイミング信号を生成するビデオタイミング信号生成手段と、

前記ビデオフレームとビデオタイミング信号を受け取り、前記ビデオフレームを前記ビデオタイミング信号に同期してデコードするビデオデコード手段とを有することを特徴とするマルチプレクスオーディオビデオ信号のデコーディングシステム。

【請求項2】 前記分離手段から前記ビデオタイムスタンプを受け取るビデオタイムスタンプ記憶手段をさらに備えることを特徴とする請求項1に記載のデコーディングシステム。

【請求項3】 前記分離手段から前記オーディオタイムスタンプを受け取るオーディオタイムスタンプ記憶手段をさらに備えることを特徴とする請求項1または2に記載のデコーディングシステム。

【請求項4】 前記分離手段から前記ビデオフレームを受け取るビデオ記憶手段をさらに備えることを特徴とする請求項1、2または3に記載のデコーディングシステム。

【請求項5】 前記分離手段から前記オーディオサンプル群を受け取るオーディオ記憶手段をさらに備えることを特徴とする請求項1乃至4のいずれかに記載のデコーディングシステム。

【請求項6】 前記タイムリファレンスが参照時刻から経過した絶対時刻を表すことを特徴とする請求項1乃至5のいずれかに記載のデコーディングシステム。

【請求項7】 前記タイムリファレンス生成手段が90

KHzで動作することを特徴とする請求項6に記載のデコーディングシステム。

【請求項8】 前記マルチプレクスオーディオビデオ信号がMPEGスタンダード信号であることを特徴とする請求項1乃至7のいずれかに記載のデコーディングシステム。

【請求項9】 前記オーディオタイミング信号生成手段が、前記オーディオタイムスタンプを、前記タイムリファレンスにより計測した場合に、前記オーディオタイムスタンプの値に最も近い時刻でデコードされたオーディオサンプル群に関連付けることを特徴とする請求項6に記載のデコーディングシステム。

【請求項10】 前記ビデオタイミング信号生成手段は、前記ビデオタイムスタンプを、前記タイムリファレンスにより計測した場合に、前記ビデオタイムスタンプの値に最も近い時刻でデコードされたビデオフレームに関連付けることを特徴とする請求項6に記載のデコーディングシステム。

【請求項11】 MPEGスタンダードマルチプレクスオーディオビデオ信号のデコーディングシステムであって、

参照時刻から経過した絶対時刻を表す90KHzのタイムリファレンスを生成するタイムリファレンス生成手段と、

オーディオタイムスタンプがオーディオサンプル群と等しいか、または少ない数であり、かつ、ビデオタイムスタンプがビデオフレームと等しいか、または少ない数である場合において、コンボジットデータストリームから、前記オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプを分離する分離手段と、

前記分離手段から前記ビデオタイムスタンプを受け取り、記憶するビデオタイムスタンプ記憶手段と、

前記分離手段から前記オーディオタイムスタンプを受け取り、記憶するオーディオタイムスタンプ記憶手段と、前記分離手段から前記ビデオフレームを受け取り、記憶するビデオ記憶手段と、

前記分離手段から前記オーディオサンプル群を受け取り、記憶するオーディオ記憶手段と、

前記オーディオタイムスタンプ記憶手段から前記オーディオタイムスタンプを受け取るとともに、前記タイムリファレンスを受け取り、前記タイムリファレンスを前記オーディオタイムスタンプと比較し、前記オーディオサンプル群の1つを前記オーディオタイムスタンプの各々に関連付けるオーディオタイミング信号を生成し、かつ、前記タイムリファレンスにより測定した場合において、前記オーディオタイムスタンプの値に最も近い時刻でデコードされた前記オーディオサンプル群に前記オーディオタイムスタンプを関連付けるオーディオフエイズロックループ手段と、

前記オーディオサンプル群とオーディオタイミング信号を受け取り、前記オーディオサンプル群を前記オーディオタイミング信号に同期してデコードするオーディオデコード手段と、

前記ビデオタイムスタンプを前記ビデオタイムスタンプ記憶手段から受け取るとともに、前記タイムリファランズを受け取り、前記タイムリファランズを前記ビデオタイムスタンプと比較し、前記ビデオフレームの1つを前記ビデオタイムスタンプの各々に関連付けるビデオタイミング信号を生成し、前記タイムリファランズにより測定した場合において、前記ビデオタイムスタンプの値に最も近い時刻でデコードされた前記ビデオフレームに前記ビデオタイムスタンプに関連付けるビデオフェイズロックループ手段と、

前記ビデオフレームとビデオタイミング信号を受け取り、前記ビデオフレームを前記ビデオタイミング信号に同期してデコードするビデオデコード手段とを備えることを特徴とするデコーディングシステム。

【請求項12】 タイムリファランズを生成するタイムリファランズ生成手段と、

データタイムスタンプがデータユニットと等しいか、または少ない数の場合において、コンボジットデータストリームを前記データユニットとデータタイムスタンプとに分離する分離手段と、

前記タイムスタンプとタイムリファランズを受け取り、前記タイムリファランズを前記タイムスタンプと比較し、前記データユニットの1つを前記タイムスタンプの各々に関連付けるタイミング信号を生成し、前記タイムリファランズにより測定した場合において、前記データタイムスタンプの値に最も近い時刻でデコードされた前記データユニットに前記タイムスタンプに関連付けるフェイズロックループ手段と、

前記データユニットとオーディオタイミング信号を受け取り、前記ユニットを前記タイミング信号に同期してデコードするデコード手段とを備えることを特徴とするデコーディングシステム。

【請求項13】 コンボジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する分離手段と、

前記オーディオタイムスタンプを受け取り、前記オーディオタイムスタンプに同期するオーディオタイミング信号を生成するオーディオフェイズロックループ手段と、前記オーディオサンプル群とオーディオタイミング信号を受け取り、前記オーディオサンプル群を前記オーディオタイミング信号に同期してデコードするオーディオデコード手段と、

前記ビデオタイムスタンプを受け取り、前記ビデオタイムスタンプに同期するビデオタイミング信号を生成するビデオフェイズロックループ手段と、

前記ビデオフレームとビデオタイミング信号を受け取り、前記ビデオフレームを前記ビデオタイミング信号に同期してデコードするビデオデコード手段とを備え、前記オーディオデコード手段は、前記オーディオサンプル群からアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積した前記アクセスユニットカウントと比較し、前記オーディオデコードが正しく同期して動作していることを確認するオーディオ確認手段を含み、

前記ビデオデコード手段は、前記ビデオフレームからアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積した前記アクセスユニットカウントと比較し、前記ビデオデコードが正しく同期して動作していることを確認するビデオ確認手段とを含むことを特徴とするマルチプレクスオーディオビデオ信号のデコーディングシステム。

【請求項14】 前記分離手段から前記ビデオタイムスタンプを受け取るビデオタイムスタンプ記憶手段をさらに備えることを特徴とする請求項13に記載のデコーディングシステム。

【請求項15】 前記分離手段から前記オーディオタイムスタンプを受け取るオーディオタイムスタンプ記憶手段をさらに備えることを特徴とする請求項13に記載のデコーディングシステム。

【請求項16】 前記分離手段から前記ビデオフレームを受け取るビデオ記憶手段をさらに備えることを特徴とする請求項13、14または15に記載のデコーディングシステム。

【請求項17】 前記分離手段から前記オーディオサンプル群を受け取るオーディオ記憶手段をさらに備えることを特徴とする請求項13乃至16のいずれかに記載のデコーディングシステム。

【請求項18】 コンボジットデータストリームからデータユニットとデータタイムスタンプとを分離する分離手段と、

前記データタイムスタンプを受け取り、前記データタイムスタンプに同期するタイミング信号を生成するタイミング信号生成手段と、

前記データユニットとタイミング信号を受け取り、前記データユニットを前記タイミング信号に同期してデコードするデコード手段とを備え、

前記デコード手段は、前記各バケットからアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積された前記アクセスユニットカウントと比較し、前記デコード手段が正しく同期して動作していることを確認する確認手段を備えることを特徴とするデコーディングシステム。

【請求項19】 前記タイミング信号生成手段は、フェイズロックループを含むことを特徴とする請求項18に記載のデコーディングシステム。

【請求項20】 マルチプレクスオーディオビデオ信号をデマルチプレクスする方法であって、コンボジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する工程と、前記オーディオタイムスタンプに同期するオーディオタイミング信号を生成する工程と、前記オーディオタイミング信号に同期して、前記オーディオサンプル群をデコードする工程と、前記オーディオサンプル群からアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積された前記アクセスユニットカウントと比較し、オーディオ同期を確認する工程と、前記ビデオタイムスタンプに同期するビデオタイミング信号を生成する工程と、前記ビデオタイミング信号に同期して、前記ビデオフレームをデコードする工程と、前記ビデオフレームからアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積された前記アクセスユニットカウントと比較し、ビデオ同期を確認する工程と、蓄積された前記アクセスユニットカウントを、新しいアクセスユニットが受け取られる毎に、インクリメントする工程とを備えることを特徴とするデマルチプレクス方法。

【請求項21】 前記ビデオフレームのアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより大きい場合、前記ビデオフレームを再現する工程をさらに備えることを特徴とする請求項20に記載のデマルチプレクス方法。

【請求項22】 前記ビデオフレームのアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより小さい場合、前記ビデオフレームを削除する工程をさらに備えることを特徴とする請求項20または21に記載のデマルチプレクス方法。

【請求項23】 前記オーディオサンプル群のアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより小さい場合、前記オーディオサンプル群を削除する工程をさらに備えることを特徴とする請求項20、21または22に記載のデマルチプレクス方法。

【請求項24】 前記オーディオサンプル群のアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより大きい場合、前記オーディオサンプル群を再現する工程をさらに備えることを特徴とする請求項20乃至23のいずれかに記載のデマルチプレクス方法。

【請求項25】 ビデオ信号と付随するオーディオ信号を処理する信号処理方法であって、オーディオサンプル群とビデオフレームを有する信号を供給する工程と、前記オーディオサンプル群をデコードする工程と、

前記オーディオサンプル群からアクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積された前記アクセスユニットカウントと比較し、オーディオの同期を確認する工程と、

前記ビデオフレームをデコードする工程と、

前記ビデオフレームから前記アクセスユニットカウントを抽出し、前記アクセスユニットカウントを、蓄積された前記アクセスユニットカウントと比較し、ビデオの同期を確認する工程と、

蓄積された前記アクセスユニットカウントを、新しいアクセスユニットが受け取られる毎に、インクリメントする工程とを備えることを特徴とする信号処理方法。

【請求項26】 前記ビデオフレームのアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより大きい場合、前記ビデオフレームを再現する工程をさらに備えることを特徴とする請求項25に記載の信号処理方法。

【請求項27】 前記ビデオフレームのアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより小さい場合、前記ビデオフレームを削除する工程をさらに備えることを特徴とする請求項25または26に記載の信号処理方法。

【請求項28】 前記オーディオサンプル群のアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより小さい場合、前記オーディオサンプル群を削除する工程をさらに備えることを特徴とする請求項25、26または27に記載の信号処理方法。

【請求項29】 前記オーディオサンプル群のアクセスユニットカウントが、蓄積された前記アクセスユニットカウントより大きい場合、前記オーディオサンプル群を再現する工程をさらに備えることを特徴とする請求項25乃至28のいずれかに記載の信号処理方法。

【請求項30】 前記オーディオビデオ信号は、前記オーディオサンプル群とビデオフレームの所定のものの表示に関連するオーディオタイムスタンプとビデオタイムスタンプとを含むことを特徴とする請求項25乃至29のいずれかに記載の信号処理方法。

【請求項31】 データビットストリームを処理する信号処理方法であって、

第1のフレームヘッダに対応するアクセスユニットカウントによりレジスタを初期化する工程と、

1以上の連続するフレームを受け取り、デコードする工程と、

連続する前記フレームの各々の前記レジスタをインクリメントし、アクセスユニットカウントを有する前記レジスタの内容を、連続する前記フレームの各々と比較する工程と、

連続する前記フレームの各々からの前記アクセスユニットカウントが、前記レジスタに蓄積した値と等しいか否かを判定し、同期を確認する工程と、

前記ビデオフレームのアクセスユニットカウントが、蓄積した前記アクセスユニットカウントより小さい場合、前記ビデオフレームを削除する工程と、不連続ビットを検出する工程と、前記不連続ビットの検出に対応して、前記レジスタを再度初期化する工程とを備えることを特徴とする信号処理方法。

【請求項32】 前記レジスタは、前記不連続ビットに続くアクセスユニットカウントにより再度初期化することを特徴とする請求項31に記載の信号処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、一般的に、オーディオおよびビデオ情報のデジタル的伝送または蓄積の分野に関する。特に本発明は、ISO (International Organization for Standardization) のMPEG (Motion Pictures Expert Group) のような、マルチプレクスされ、かつバック化されたオーディオおよび、またはビデオ (A/V) 情報のタイミングを確認する技術に関する。

【0002】

【従来の技術】ここにおいて用いられるアクセスユニットの言葉は、1フレームのビデオデータまたは1バッチ (batch) (群) のオーディオサンプル、若しくは1群の他のデータサンプルを意味する。一般的に、デコーダは、所定の周期で、このアクセスユニットを処理し、デコードされたアクセスユニットを出力する。この周期は、例えばビデオの場合、ピクチャレートであり、オーディオの場合、オーディオサンプリングレートの整数倍 (オーディオアクセスユニットにおけるオーディオサンプルの数に等しい) となる。

【0003】例えば、ISOドキュメント番号1-11172に規定されているようなISOスタンダードMPEGシステムアプリケーションにおいては、アクセスユニットのデコーディング時刻を表すタイムスタンプ (ATS, VTS) が、図7に示すように、マルチプレクスされたビデオおよびオーディオパケットのマルチプレクスシンタックスの中に含まれている。これらのタイムスタンプは、マルチプレクスシンタックス、即ち、パケットヘッダの中に含まれている。

【0004】タイムスタンプは、そのパケット内の最初のアクセスユニットヘッダのデコーディング時刻を表している。各パケットは、複数のアクセスユニットを含むことができるため、全てのアクセスユニットがタイムスタンプに関連付けられているわけではない。

【0005】マルチプレクスアプリケーションは、複数のデコーダを用いるが、ISOのMPEG1システムスタンダードは、1つのデコーダのみを有するアプリケーションにもまた、適用が可能である。固定されたビット

レートの (例えばオーディオ) アプリケーションにおいて、完全なるクロックは存在しない。従って、デジタルストレージメディア (DSM) ビットレート (または伝送ビットレート)、およびデコーダ入力ビットレートは、例えばサンプリングクロック周波数エラーによって変化する。

【0006】このようなシステムにおいて、デコーダは、一般的に、一度に1つのアクセスユニットを読み込む。さらに、クロックの差異に起因して、伝送ビットレート (記録媒体より再生出力されるデータのビットレート) と、デコーダへ入力されるデータのビットレートは一致しない。このような差異を補償するために、バッファを使用することが可能である。

【0007】ビデオデータの場合、連続するアクセスユニット (即ち、フレーム) は、ピクチャの内容に対応して圧縮され、可変長符号化 (VLC) されたアクセスユニットとなる。その結果、ビデオデコーダ入力ビットレートは、大きなばらつきを有し、比較的大きなバッファを必要とする。

【0008】しかしながら、バッファを用いたとしても、そもそも理想的なビットレートは存在し得ず、その結果、バッファエラー (バッファのオーバフローまたはアンダフロー) が起こり得る。バッファのオーバフローとアンダフローを防止するために、一般的に2つの方法が用いられる。

【0009】デジタルストレージメディアスレーブと称される1つの方法においては、伝送レート (即ち、バッファ入力レート) が制御される。

【0010】デコーダスレーブと称される他の方法によれば、バッファ出力のデータレートが制御される。ビデオの場合、これは、フレームレートを調整することにより実行される。オーディオの場合、サンプリングレートを調整することにより実行される。バッファ出力データレートは、このように調整される。

【0011】他のデコーダスレーブ方法は、バッファ出力データレートを制御するために、アクセスユニットをスキップするか (オーバフローの恐れがあるとき)、あるいは繰り返し出力する (アンダフローの恐れがあるとき)。

【0012】デコーダレートの調整および伝送ビットレートの調整は、周辺ハードウェアの特性によって制限される。従って、バッファエラー (即ち、理想的なバッファの蓄積状態からの偏差) があまりにも大きすぎる場合、適切な制御が困難になるか、あるいは不可能になる。大きなバッファエラーは、再生開始時に、ときどき発生する。従って、一般的に、デコーダは初期のバッファエラーを抑制するために、適当なスタートアップディレイを与えた後に再生を開始する。

【0013】MPEG1システムスタンダードにおいては、デコーダレートまたは伝送レートを制御するために

使用可能なフィールドが、マルチプレクスシンタックスに含まれている。バックヘッダのSCR (System Clock Reference) と称される値を、伝送データレートの制御に用いることができる。ビデオパケットヘッダのタイムスタンプは、フレームレートを制御するために用いることができ、オーディオパケットヘッダのタイムスタンプは、サンプリングレートを制御するために用いることができる。SCRは、パケットデータ (の先頭部分) がデコーダバッファに入力される時刻を表し、タイムスタンプは、パケットデータ内の所定のアクセスユニットがデコーダバッファから出力される時刻 (デコーディングされる時刻) を表している。

【0014】SCRとタイムスタンプは、両方とも90 KHzのレートで連続的にインクリメントするクロックの絶対値で表されている。従って、最初に読み出されたSCRと、最初に読み出されたタイムスタンプの差は、スタートアップディレイとして用いることができる。

【0015】残念ながら (最初のスタートアップのための最初のオーディオタイムスタンプとビデオタイムスタンプを除き)、このタイムスタンプを連続的に使用することは困難である。なぜならば、デマルチプレクスの後、タイムスタンプが関係するアクセスユニットから分離されてしまうからである。即ち、デコーディングシステムのデマルチプレクススイッチが、パケットデータからタイムスタンプを分離し、そのパケットデータを (タイムスタンプを付随させることなく) それぞれのバッファ内に蓄積するため、どのタイムスタンプが、どのアクセスユニットヘッダに属するのかを追跡することが困難になる。

【0016】例えば、あるデコーディングシステムが所定のMPEGマルチプレクスストリームを処理する。最初のSCR (System Clock Reference) が検出され、システムがこの値をローカルクロック (90 KHz) の初期化のために用いる。そのときから、このクロックは自動的に90 KHzのレートでインクリメントされる。そして、システムは、最初のビデオDTS (Decoding Time Stamps) を検出する。このDTSは、90 KHzのクロックの値で、最初のこの後に続くピクチャヘッダのデコーディングタイムを表している。

【0017】しかしながら、一般的に、このピクチャヘッダは、デコードされる前に、ビデオデコーダの前のバッファに起因して、相当遅延しており、従って、使用する前に、タイムスタンプも遅延するか、あるいは記憶しておく必要がある。さらに、最初のピクチャがデコードされる前に、さらにいくつかのタイムスタンプを有するビデオパケットが、デマルチプレクススイッチに到来する。従って、これらのタイムスタンプも、使用する前に遅延されるか、どこかに保持されなければならない。

【0018】パケットヘッダは、1つのタイムスタンプ

のみを有し、パケットデータは、いくつかのアクセスユニットヘッダを有することができるので、全てのアクセスユニットヘッダがタイムスタンプを有する訳ではない。パケットデータが1つのアクセスユニットヘッダを有するものとしても、先行するパケットヘッダにタイムスタンプを挿入することは選択的なことであり、義務ではない。なぜならば、タイムスタンプに関し、MPEGで要求していることは、タイムスタンプを少なくとも0.7秒毎に1回発生させることだけであるからである。

【0019】仮に、特定のアプリケーションが2以上のデコーダを (例えば、1つのビデオデコーダと、1つのオーディオデコーダを) 使用するものとする、これらのデコーダを同期させることが度々必要になる。MPEG1システムスタンダードによれば、タイムスタンプは、この同期化を実現するのに使用されなければならない。このスタンダードは、オーディオデコーダの処理による遅延と、ビデオデコーダの処理による遅延を、0 msと仮定している。その結果 (このモデルにおいては)、アクセスユニットがデコードされる時刻は、デコードされたアクセスユニット (即ち、プレゼンテーションユニット) が出力される時刻である。

【0020】デコーダの (初期バッファエラーを最小化するための) スタートアップディレイについて、上述した理由の他、マルチデコーダシステムにおいては、各デコーダ毎にイニシャルスタートアップディレイが必要となる。これは、例えば、 (同じパック内において) 同時に発生するオーディオとビデオのパケットが、デコードされた後、異なる時刻に出力されるべきオーディオとビデオ信号のセグメントであるようなことが、時折、起きるからである。

【0021】これは、有り得ることである。なぜならば、MPEGは、各ビデオデコーダとオーディオデコーダの前段における所定量の記憶を認めており、これにより、フレキシブルなマルチプレクスビットストリーム構造が許容されるからである。MPEGマルチプレクススタンダードを記述する他の方法は、そのデコーダにおけるバッファが、オーバフローあるいはアンダフローしない限り、如何なる種類のバックおよびパケット構造も受け入れられるとしている。

【0022】同期化のためのデコーディングシステムとして、2つの種類が存在する。第1はロックシステムであり、そこにおいては、フレームレートとサンプリングレートが1つの共通のクロック (例えば、MPEGにおける90 KHz) にロックされている。このロックシステムは、フレームレートとサンプリングレートが同様に1つの共通のクロックにロックするようになされたエンコードシステムにおいて生成されたビットストリームしか再生することができないという欠点を有している。ビデオエンコーダのクロックとオーディオエンコーダのク

ロックがロックしているか否かは、アプリケーションによる(CDI-FMV(即ち、MPEG1スタンダードを採用するフルモーションビデオエクステンションを有する、フィリップス社のコンパクトディスクインターラクティブ)の場合においては、エンコーダをロックすることが義務付けられている)。

【0023】このシステムにおいては、(仮に、伝送エラー特性が制限されていれば)最初のビデオタイムスタンプとオーディオタイムスタンプを読み取り、それをそれぞれのデコードの開始のために使用した後、それに続く全てのタイムスタンプは、無視することができる。この種のデコーディングシステムは、比較的シンプルであり、どのタイムスタンプが、どのアクセスユニットに属するのかを追跡する必要がない。しかしながら、伝送エラーが、アクセスユニットヘッダを消失させたり、誤って発生させれば、シンクエラー(およびそれに対応するバッファエラー)が起きる。このような問題は、MPEGシステム以外のシステムにおいても発生する。本発明は、この問題を解決するものである。

【0024】第2の種類のデコーディングシステム(ノンロックデコーディングシステムと称される)は、また、ノンロックエンコードマルチプレクスビットストリームを再生することができる。ノンロックエンコードビットストリームは、独立したエンコーダフレームレートクロックとサンプリングレートクロックを有するエンコードシステムによって発生されたものである。この場合、ビデオエンコーダのフレームレートエラーと、オーディオエンコーダのサンプリングレートエラーは無関係であり、それらは独立に変化するものである。

【0025】ノンロックMPEGデコーディングシステムは、マルチプレクスビットストリームが、ノンロックエンコーディングシステム、即ち、ピクチャレートクロックとオーディオサンプリングレートクロックが独立しているエンコードシステムによって生成された場合において使用される。ノンロックエンコードシステムが使用されるか否かは、アプリケーションに依存する。

【0026】例えば、CDI-FMVの場合、独立したビデオエンコーダクロックとオーディオエンコーダクロックが許容される。その代わり、これらの両方のクロックが1つの共通のクロックとロックする必要がある。しかしながら、将来、ノンロックMPEGシステムを使用するアプリケーションが現れるかもしれない。

【0027】MPEGスタンダードが使用されたとき、ノンロックエンコーダ(フレームレートおよびサンプリングレート)クロックエラーは、タイムスタンプとともに記録され、ビットストリームに含まれる。再生の間、AVシンクエラーを防止するため、少なくとも1つのデコーダがPLL機構を有し、そのPLL機構が周期的にタイムスタンプを使用し、実際のフレーム(またはサンプリング)レートをタイムスタンプ値と一致させる必要

がある。

【0028】ビデオデコーダは、このように、ビデオタイムスタンプ(即ち、ビデオDTSまたはビデオPTS(Presentation Time Stamps))を読み出し、これらのタイムスタンプを、ピクチャレートを制御するために使用しなければならない。あるいはまた、オーディオデコーダが、オーディオDTSを読み取り、これらを、オーディオサンプリングレートを制御するために使用しなければならない。あるいはまた、両方のデコーダが、これらのタイムスタンプを、それらのクロックを制御するために用いなければならない。

【0029】

【発明が解決しようとする課題】図7において、MPEGまたは同様のパケットのマルチプレクスビットストリーム20は、ビデオパケット22とオーディオパケット24の混合として示されている。パケット22とパケット24の集合が、バックヘッダ26により先行される、より大きなバックとされている。それぞれの場合において、実際のビデオデータ27またはオーディオデータ28は、(他のデータ項目とともに)タイムスタンプ(VTS)30を有するビデオパケットヘッダ、または(他のデータ項目とともに)タイムスタンプ(ATS)32を有するオーディオパケットヘッダにより、それぞれ先行されている。実際のビデオデータ27は、ビデオフレームに分割され、オーディオデータ28は、図示されているように、サンプルの群(ユニット)に分割される。

【0030】1つのデコーディング方法によれば、デコーディングシステムが、入力パケットを、オーディオビットストリームとビデオビットストリームにデマルチプレクスし、そのパケットヘッダからタイムスタンプを取り出し、このタイムスタンプを、各エレメンタリストリーム(ビデオビットストリームまたはオーディオビットストリーム)の関係するアクセスユニットの直前に挿入する。

【0031】これにより、図8に示すようなシンタックスが生成される。このシンタックスにおいて、例えばフレーム $n+1$ は、ビデオタイムスタンプ(VTS)(90KHzクロックによる時間値)を有している。同様に、オーディオユニット $m+1$ は、その直前にオーディオタイムスタンプ(ATS)を含んでいる。

【0032】このようなエレメンタリストリームを生成するために、図9に示すようなデコーディングシステムを用いることができる。このシステムにおいては、マルチプレクスビットストリーム20が、デマルチプレクサ50に供給され、このデマルチプレクサ50は、ビットストリームを、ビデオデータ、ビデオタイムスタンプ、オーディオデータおよびオーディオタイムスタンプに分離する。

【0033】ビデオデータは、ビデオシンタックス変更

回路54に供給され、オーディオデータは、オーディオシンタックス変更回路58に供給される。そこで、シンタックスが図8に示すように変更されたビデオデータとオーディオデータは、ビデオデコーダ66とオーディオデコーダ68において、それぞれピクチャレートコントロール回路74とサンプルレートコントロール回路76の制御の下、デコーディングされる前にバッファ60と62に記憶される。ビデオとオーディオは、出力84と86にそれぞれ現れる。

【0034】この方法は、デマルチプレクスされたデータストリームの適正な位置にタイムスタンプを挿入するために、各バッファ60、62の前に特別のモジュール(変更回路54、58)が必要になるという欠点を有している。

【0035】このノンロックシステムは、さらに、エレメンタリ(A/V)ビットストリームの両方が変更されるという欠点を有している(即ち、デコーダ66、68の入力におけるビットストリームは、それぞれオーディオスタンダードとビデオスタンダードに従わないことになる)。従って、このシステムのデコーダは、ノンマルチプレクスオーディオビットストリームとビデオビットストリームを直接デコードすることができない。

【0036】結局、このノンロックシステムは、タイムスタンプとアクセスユニットヘッダとの間の関係を保持することができる。従って、(例えば、伝送エラーにより)アクセスユニットヘッダが失われたか、あるいは、誤って生成されたかを検出する能力を有する。このようなエラーは、意図するアクセスユニットのデコード時刻(即ち、タイムスタンプの値)と、作用を受けたデコーダのPLLによって検出された実際のアクセスユニットのデコード時刻との間に大きな差をもたらす原因となる。

【0037】しかしながら、残念なことに、PLLは、デコーダのクロックを調整して、このような差を修正するよう設計されており、これにより、最悪の場合、第3の欠点が発生する。PLLは、デコーダレートを(通常)非常にゆっくり調整して、大きなAVシンクエラーを修正しようとする。このようなゆっくりした修正処理は、AVシンクエラーとそれに対応するバッファエラーを修正するのに、長い時間を必要とする。その結果、バッファがアンダフローあるいはオーバフローする可能性が増加する。また、大きなAVシンクエラーは、数秒あるいはそれ以上継続するので、ユーザがAVシンクエラーに気付く可能性も増加する。

【0038】PLLが作用を受けたデコーダクロックを迅速に調整することによって、AVシンクエラーを修正しようとするれば、オーディオまたはビデオに他の影響が現れる(例えば、ビデオの垂直ロールまたはオーディオにおける周波数シフト)。従って、このシステムにおいて、タイムスタンプ値によりデコーダクロックを調整す

ることは、ノンロックエンコードコンポジットビットストリームに存在する小さなエンコーダクロックエラーを再現するのに適切である。このアプローチは、所定の数のアクセスユニットヘッダが失われたり、過剰となったりすることによって発生する大きなシンクエラーを修正するには、常に適切なものという訳ではない。

【0039】最も単純なMPEGデコーディングシステム、および最も単純なノンMPEGデコーディングシステムは、ロックされたデコーディングシステムである

(ビデオデコーダのピクチャレートとオーディオデコーダのサンプリングレートが1つの共通のクロックにロックしたシステムである)。これらのシステムは、(例えば、伝送エラーや記憶エラーに起因して)アクセスユニットヘッダを失ったり、あるいは誤って生成したアプリケーションにおいて使用することができない欠点を有している。

【0040】本発明の目的は、アクセスユニットヘッダの消失または過剰を検出することができるデコーディングシステムを提供することにある。

【0041】本発明の他の目的は、アンロックシステムにおいて、同期化を維持することができるデコーディングシステムを提供することにある。

【0042】本発明のさらに他の目的は、編集されたマルチプレクスビットストリームおよび編集されたエレメンタリビットストリームを再生することができるデコーディングシステムを提供することにある。

【0043】

【課題を解決するための手段】本発明の一実施例においては、マルチプレクスされたオーディオビデオ信号のためのデコーディングシステムが、タイムリファランスを生成するクロックを有している。デマルチプレクススイッチは、オーディオタイムスタンプが、オーディオサンプル群の数と等しいか、それより少なく、また、ビデオタイムスタンプが、ビデオフレームの数と等しいか、それより少ない場合において、コンポジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する。オーディオフェイズロックループは、オーディオタイムスタンプとタイムリファランズを受け取り、タイムリファランズとオーディオタイムスタンプとを比較し、オーディオサンプル群の1つを各オーディオタイムスタンプに関連付けるオーディオタイミング信号を生成する。オーディオデコーダは、オーディオサンプル群とオーディオタイミング信号とを受け取り、オーディオタイミング信号に同期して、オーディオサンプル群をデコードする。ビデオフェイズロックループは、ビデオタイムスタンプとタイムリファランズを受け取り、タイムリファランズとビデオタイムスタンプとを比較し、ビデオフレームの1つを各ビデオタイムスタンプに関連付けるビデオタイミング信号を生成する。ビデオデコーダは、

ビデオフレームとビデオタイミング信号とを受け取り、ビデオタイミング信号に同期して、ビデオフレームをデコードする。

【0044】MPEGスタンダードマルチプレクスオーディオビデオ信号のためのデコーディングシステムは、参照時刻から経過した絶対時刻を表すタイムリファランスを生成する、90KHzのクロックを含む。デマルチプレクススイッチは、オーディオタイムスタンプが、オーディオサンプル群の数と等しいか、それより少ない数であり、ビデオタイムスタンプが、ビデオフレームの数と等しいか、それより少ない数である場合において、コンポジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する。ビデオタイムスタンプバッファは、デマルチプレクススイッチから、ビデオタイムスタンプを受け取り、オーディオタイムスタンプバッファは、デマルチプレクススイッチから、オーディオタイムスタンプを受け取る。ビデオバッファは、デマルチプレクススイッチからビデオフレームを受け取り、オーディオバッファは、デマルチプレクススイッチからオーディオサンプル群を受け取る。オーディオフィーズロックループは、オーディオタイムスタンプバッファからオーディオタイムスタンプを受け取り、また、タイムリファランスを受け取る。オーディオPLLは、タイムリファランスとオーディオタイムスタンプを比較し、オーディオサンプル群の1つを各オーディオタイムスタンプに関連付けるオーディオタイミング信号を生成する。オーディオフィーズロックループは、クロックを計測することで、オーディオタイムスタンプの値に最も近い時刻においてデコードされたオーディオサンプル群に、オーディオタイムスタンプに関連付ける。オーディオデコーダは、オーディオサンプル群とオーディオタイミング信号を受け取り、オーディオタイミング信号に同期して、オーディオサンプル群をデコードする。ビデオフェイズロックループは、ビデオタイムスタンプバッファからビデオタイムスタンプを受け取り、また、タイムリファランスを受け取る。ビデオPLLは、タイムリファランスとビデオタイムスタンプとを比較し、各ビデオタイムスタンプをビデオフレームの1つに関連付けるビデオタイミング信号を生成する。この場合、ビデオフェイズロックループは、クロックを計測し、ビデオタイムスタンプの値に最も近い時刻においてデコードされたビデオフレームとビデオタイムスタンプとを関連付ける。ビデオデコーダは、ビデオフレームとビデオタイミング信号を受け取り、ビデオタイミング信号に同期して、ビデオフレームをデコードする。

【0045】本発明の他の面においては、デコーダがタイムリファランスを生成するクロックを含んでいる。コンポジットデータストリームは、データタイムスタンプが、データユニットと等しいか、少ない数となっている

場合において、データユニットとデータタイムスタンプとに分離される。フェイズロックループは、タイムスタンプとタイムリファランスを受け取り、タイムリファランスをタイムスタンプと比較し、各タイムスタンプをデータユニットの1つに関連付けるタイミング信号を生成する。この場合、フェイズロックループはクロックを計測し、ビデオタイムスタンプの値に最も近い時刻においてデコードされたデータユニットとタイムスタンプを関連付ける。デコーダは、データユニットとオーディオタイミング信号を受け取り、タイミング信号に同期して、ユニットをデコードする。

【0046】本発明の他の面においては、マルチプレクスされたオーディオビデオ信号用のデマルチプレクサが、コンポジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離する、デマルチプレクススイッチを有する。オーディオフィーズロックループは、オーディオタイムスタンプを受け取り、オーディオタイムスタンプに同期したオーディオタイミング信号を生成する。オーディオデコーダは、オーディオサンプル群およびオーディオタイミング信号を受け取り、オーディオタイミング信号に同期して、オーディオサンプル群をデコードする。オーディオデコーダは、オーディオサンプル群から、アクセスユニットカウントを抽出し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、オーディオデコーダが正しく同期して動作しているか否かを確認する。ビデオフェイズロックループは、ビデオタイムスタンプを受け取り、ビデオタイムスタンプに同期するビデオタイミング信号を生成する。ビデオデコーダは、ビデオフレームとビデオタイミング信号を受け取り、ビデオフレームをビデオタイミング信号に同期してデコードする。ビデオデコーダは、ビデオフレームからアクセスユニットカウントを抽出し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、ビデオデコーダが正しく同期して動作しているか否かを確認する。

【0047】本発明の他の面においては、デコーダが、コンポジットデータストリームをデータユニットとデータタイムスタンプとに分離する、デマルチプレクス回路を有している。タイミング回路は、データタイムスタンプを受け取り、タイムスタンプに同期してタイミング信号を生成する。デコーダは、データユニットとタイミング信号を受け取り、データユニットをタイミング信号に同期してデコードする。デコーダは、各データユニットから、アクセスユニットカウントを抽出する回路を有し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、デコーダが正しく同期して動作していることを確認する。

【0048】本発明によるマルチプレクスされたオーディオビデオ信号をデマルチプレクスする方法は、コンポ

ジットデータストリームを、オーディオサンプル群、オーディオタイムスタンプ、ビデオフレームおよびビデオタイムスタンプに分離するステップと、オーディオタイムスタンプに同期して、オーディオタイミング信号を生成するステップと、オーディオタイミング信号に同期して、オーディオサンプル群をデコードするステップと、オーディオサンプル群からアクセスユニットカウントを抽出し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、オーディオの同期化を確認するステップと、ビデオタイムスタンプに同期し、ビデオタイミング信号を生成するステップと、ビデオタイミング信号に同期して、ビデオフレームをデコードするステップと、ビデオフレームからアクセスユニットカウントを抽出し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、ビデオの同期を確認するステップと、蓄積されたアクセスユニットカウントを、新しいアクセスユニットが受け取られる毎にインクリメントするステップとを含む。

【0049】本発明のビデオおよびそれに付随するオーディオ信号を処理する方法は、オーディオサンプル群とビデオフレームを含む信号を供給するステップと、オーディオサンプル群をデコードするステップと、オーディオサンプル群からアクセスユニットカウントを抽出し、アクセスユニットカウントを、蓄積されたアクセスユニットカウントと比較し、オーディオの同期を確認するステップと、ビデオフレームをデコードするステップと、ビデオフレームからアクセスユニットカウントを抽出し、アクセスユニットカウントと、蓄積されたアクセスユニットカウントと比較し、ビデオの同期を確認するステップと、蓄積されたアクセスユニットカウントを、新しいアクセスユニットが受け取られる毎にインクリメントするステップとを有する。

【0050】データビットのストリームを処理する本発明の他の方法は、第1のフレームヘッダに対応するアクセスユニットカウントを有するレジスタを初期化するステップと、1以上の連続するフレームを受け取り、デコードするステップと、各連続するフレームのレジスタをインクリメントし、各連続フレーム毎に、レジスタの内容とアクセスユニットカウントとを比較するステップと、各連続フレームからのアクセスユニットカウントが、レジスタに蓄積されているアクセスユニットカウントと同一であるか否かを決定し、同期を確認するステップと、ビデオフレームのアクセスユニットカウントが、レジスタに蓄積されているアクセスユニットカウントより小さい場合、ビデオフレームを削除するステップと、オーディオサンプル群のアクセスユニットカウントが、蓄積されたアクセスユニットカウントより小さい場合、オーディオサンプル群を削除するステップと、不連続ビットを検出するステップと、不連続ビットを検出したとき、レジスタを再度チェックするステップとを有する。

【0051】デコーダにおけるAV同期エラーを検出し、修正するために、(デジタルAVストレージメディアアプリケーションまたはデジタルAV伝送アプリケーションのための)ISOのMPEG1システムスタンダードによれば、タイムスタンプは、マルチプレクスシンタックスのバケットヘッダに含まれている。タイムスタンプは、そのバケットにおいて発生する最初のアクセスユニットヘッダ(即ち、フレームヘッダまたはオーディオユニットヘッダ)の表示時刻を表す。デマルチプレクスされた後、タイムスタンプは、通常、関係するアクセスユニットから分離される。これが、その使用を困難にする。例えば、ノンロックデコーディングシステムにおいては、どのタイムスタンプが、どのアクセスユニットに属するかを発見する何らかの方法が採用されなければならない。本発明の一実施例によれば、比較的単純なデコーディングシステムが、この関係を決定するタイムスタンプ値の冗長性を利用する。本発明は、各タイムスタンプの値を見て、それを所定のアクセスユニットに関連付けることによって、どのタイムスタンプが、どのアクセスユニットに属するかを決定する問題を解決する。これは、ノンロックデコーディングを容易にする。

【0052】本発明の実施例による圧縮ビデオ信号および圧縮オーディオ信号デコーディング装置は、アクセスユニットヘッダ(例えば、フレームヘッダ、オーディオユニットヘッダまたはその他のデータユニットヘッダ)が、ストレージメディアまたは伝送エラーにより失われたり、あるいは生成された場合のためのデジタルAVアプリケーション用のローカルアクセスユニットカウントを含む。AVシンクエラーの発生を抑制するために、アクセスユニットカウントと称される値が、各エレメンタリストリームに含まれる。この値により、各デコーダは、アクセスユニットヘッダの消失、あるいは誤りを検出することができ、AV同期を修正することができる。編集を支持するために、不連続性を表す特別なビットを付加することができる。

【0053】アクセスユニットカウントフィールドが各エンコーダの各エレメンタリビットストリームのシンタックスに含まれる実施例においては、このデータをマルチプレクスレアーから各エレメンタリに移動する必要がない。従って、デコーディングシステムのハードウェアあるいはソフトウェアは複雑にならない。このデコーディングシステムにおいては、タイムスタンプがイニシャルスタートアップディレイと同期に用いられ、また、エンコーダのフレームおよび、またはオーディオサンプリングレートクロックエラーの追跡に用いられる。アクセスユニットカウントは、アクセスユニットヘッダの消失または偽のアクセスユニットヘッダの検出にのみ用いられる。また、より容易なMPEGロックデコーダシステムが可能であり、本発明は、ノンMPEG装置においても使用が可能である。

【0054】

【作用】各エレメンタリストリームにおいて、アクセスユニットカウント値を使用することで、各デコーダは、アクセスユニットヘッダの消失や誤ったアクセスユニットを検出することができる。作用を受けたデコーダは、システムに対し、アクセスユニットを再度デコーディングすることによって、あるいはまた、アクセスユニットをスキップすることによって、同期化エラーの修正やリセットを要求することができる。

【0055】このようなシステムのデコーダは、容易に提供することができる。加えて、アクセスカウントは、上記したノンロックデコーディングシステムの欠点を有しない、よりシンプルなノンロックデコーディングシステムを可能にする。

【0056】

【実施例】上記した問題のうちのいくつかの部分的な解決は、図1と図2に関して記述される。この解決は、MPEG1スタンダードに厳格に従うことを要求するシステムに適用することができる。このシステムにおいては、比較的シンプルなデコーディングシステムが、タイムスタンプとそれに関連付けられるアクセスユニットとの間の関係を決定するタイムスタンプ値の冗長性を利用する。

【0057】ノンロックエンコードビットストリームにおいては、例えば、ビデオタイムスタンプが、ビデオエンコーダのピクチャレート（例えば30Hz）、およびビデオエンコーダのクロックエラーに依存する、それらのピクチャのデコーディング時刻を表している。

【0058】この種のビットストリームにおいては、ビデオエンコーダのピクチャレートクロックエラーが0であり、かつ、そのエンコーディングシステムにおいては、連続するタイムスタンプの間の差は、 n を1以上の整数とすると、 $3000n$ となる（3000は、ピクチャレートが正確に30Hzであるとした場合の、各ピクチャ周期内の90KHzのサイクルの数である）。

【0059】しかしながら、クロックエラーがあると、この差は、例えば $3000n+1$ または $3000n-1$ となる（MPEGスタンダードは、エンコーダにおけるクロックエラーの限界（ $<100\text{ppm}$ ）を規定するとともに、2つの連続するタイムスタンプの間の間隔に限界（ $<0.8\text{s}$ ）を規定している）。

【0060】従って、各タイムスタンプの値を見て、かつ、処理されたピクチャの数をカウントすることにより、ビデオデコーダ（のモジュール）は、どのタイムスタンプが、どのピクチャに属するのを見いだすことができ、そして、ピクチャレートを制御するために、そのタイムスタンプを利用することができる。同様の方法は、必要であれば、オーディオデコーダにも適用することが可能である。

【0061】この部分的な解決法によるデコーディングシステムが、図1に示されている。このデコーディングシステムにおいては、マルチプレクスビットストリーム20が、デジタルシグナルプロセッサ（DSP）98の制御の下に動作する、デマルチプレクサ50に入力され、そこでビデオデータなどが分離される。分離されたビデオデータが、ビデオバッファ100に供給される。分離されたオーディオデータは、オーディオバッファ102に供給される。分離されたビデオタイムスタンプは、ビデオタイムスタンプバッファ104に供給され、分離されたオーディオタイムスタンプは、オーディオタイムスタンプバッファ108に供給される。

【0062】このシステムにおいては、各PLLがタイムスタンプを特定のアクセスユニットに関係付ける。即ち、ビデオフェイズロックループ（ビデオPLL）120とオーディオPLL122は、タイムスタンプを、そのタイムスタンプ値に最も近い位置でデコードされたアクセスユニットに関係付ける。そして、そのタイムスタンプは、オーディオデコーダ130またはビデオデコーダ128の時間設定の制御に使用される。

【0063】初期化のために、最初のバックヘッダのSCRが、クロック118（MPEG1の場合、90KHz）によって駆動される絶対時刻訂正レジスタ114に供給される。この絶対時刻の値は、ビデオPLL120およびオーディオPLL122を駆動するために用いられる。ビデオおよびオーディオPLL120および122は、それぞれビデオおよびオーディオデータを、ビデオバッファ100およびオーディオバッファ102から受け取るビデオデコーダ128およびオーディオデコーダ130の時間設定を行う。

【0064】このデコーディングシステムの動作は、図2に示されている。時刻T1において、DSP98は最初のバックヘッダを検出し、デマルチプレクサ50を位置aに切り換え、レジスタ114をSCR_aで初期化する。その後、レジスタ114の内容は、90Hzのクロック118により、1つづつインクリメントされる。時刻T2において、DSP98は最初のビデオパケットを検出し（PTSを無視し）、デマルチプレクサ50を位置cに切り換える。これにより、DTS_aが、ビデオタイムスタンプバッファ104に蓄積される。そして、次のバックヘッダが時刻T3で検出されたとき、そのヘッダのSCR_{n+1}とレジスタ114の内容との差が、DSMビットレート（マルチプレクスビットストリーム20のレート）を制御するために、図示せぬ前段の装置で用いられる。

【0065】時刻T4において、DSP98はビデオパケットを検出し、デマルチプレクサ50を位置cに切り換え、タイムスタンプDTS_{n+1}を、ビデオタイムスタンプバッファ104に蓄積させる。次に、デマルチプレクサ50は位置bにセットされ、ビデオデータをビデオ

バッファ100に供給する。時刻T5において、DSP98はデマルチプレクサ50を位置cにセットし、次のタイムスタンプDTS_{n+2}がビデオタイムスタンプバッファ104に記憶される。そして、デマルチプレクサ50は位置bにセットされ、再びデータをビデオバッファ100に供給する。

【0066】時刻T6において、レジスタ114の内容（クロック118の出力によりインクリメントされている）が、ビデオタイムスタンプバッファ104の最初のタイムスタンプDTS_nと同じになる。PLL120は、この一致を検出し、ビデオデコーダ128のフレームiのデコードをスタートさせる。時刻T7において、DSP98はオーディオ packets を検出し、デマルチプレクサ50を位置eにセットさせ、タイムスタンプDTS_nを、オーディオタイムスタンプバッファ108に記憶させる。

【0067】時刻T8において、レジスタ114の内容が、オーディオタイムスタンプバッファ108のDTS_nと同じになる。PLL122は、これを検出し、オーディオデコーダ130のユニット0のデコードをスタートさせる。

【0068】また、時刻T6（DTS_n）から1フレーム分の時間を経過したときの時刻T8において、PLL120は、ビデオデコーダ128にクロックを出力し、ビデオデコーダ128にフレームi+1のデコードを開始させる。このときタイミング（図2において、太い線で示すタイミング）は、エンコーダ側で予定しているタイミング（図2において、細い線で示すタイミング）より、何らかの理由により、若干遅くなっている。

【0069】さらに、PLL120は、次のタイムスタンプ（DTS_{n+1}）と、このタイミング時におけるレジスタ114の値との差から、このタイミングがDTS_{n+1}に対応するものであるか否かを判定する。両者の差は比較的大きいので、両者が対応していないことが判る。即ち、PLL120は、このフレーム（i+1）がタイムスタンプを有していないことを知る。

【0070】さらに、1フレーム分の時間を経過したとき、PLL120はクロックを出力し、ビデオデコーダ128がフレームi+2をデコードする。また、時刻T9において、PLL120は、このデコードのタイミングと、時刻DTS_{n+1}との差を検出し、その差が所定値以下であれば、両者を関連付ける。そして、その差から、例えば、フレームレートが遅いことを検出し、PLL120をスピードアップさせようとする。

【0071】この方法は、伝送（またはDSM）エラー特性が制限されているとき（余り大きくないとき）、うまく動作する。しかしながら、例えば伝送エラーがアクセスユニットヘッダ（例えば、図2におけるフレームヘッダi+3）を破壊した場合、この方法は、次のような問題を有している。

【0072】即ち、この例においては、時刻T9で（正確には、それより若干後のタイミングで）、ビデオデコーダ128が損傷したフレームヘッダi+3を検出できず、次のフレームヘッダi+4まで、全ての後続するデータを、単に破棄することになる。そして、次の時刻T10（正確には、それより若干後の時刻）において、ビデオデコーダ128は、フレームi+4を1ピクチャ周期だけ早く処理し（フレームi+3として処理しようとするため）、これにより、略1ピクチャ周期に等しいAVシンクエラーが発生する。

【0073】また、時刻T10において、PLL120は、タイムスタンプDTS_{n+2}を、時刻DTS_{n+2}の近くでデコードしたフレームに関係付けるが、このフレームは、伝送エラーに起因して、フレームi+4となる。従って、再びPLL120は、フレームレートが遅すぎるものと判定し、フレームレートクロックをスピードアップしようとする。しかしながら、損傷を受けたフレームヘッダi+3に起因して、フレームレートは実際には速すぎるのである。

【0074】MPEGビデオシンタックスの各フレームヘッダの現在のテンポラルリファレンスフィールドには、ピクチャ1枚につき、1ずつカウントアップするデータが記録されるため、これを用いて、フレームヘッダの消失または過剰を検出することが考えられる。しかしながら、残念なことに、この値は、各グループオブピクチャ（GOP）の最初のフレームにおいて0にリセットされる。その結果、各GOPの最後のフレームは保護されず、各GOPが1フレームだけを有している場合（例えば、全てのテンポラルリファレンスが0である場合）、ビデオビットストリームが全く保護されない。

【0075】他の選択は、MPEGビデオシンタックスのGOPヘッダに存在するタイムコードを使用することである。しかしながら、この方法は、GOPヘッダを受け取ったときにおいてのみ、エラー検出が実行可能であるという欠点を有している。多くのフレームを有するGOPの場合、シンクエラーとバッファエラーが長く続くという欠点を有している。さらに、シンクエラーが訂正されない場合、それが起こってすぐバッファがオーバフローあるいはアンダフローする可能性が増加する。

【0076】次に、図3において、アクセスユニットカウントを含むように変更された本発明のデータストリームを示す。このデータストリームにおいては、ビデオデータ27がフレームn、n+1として構成されている。本発明によれば、ビデオの各フレームは、そのフレームに先行して、アクセスユニットカウント190を有している。アクセスユニットカウント190は、例えばフレームn+1に対応し、値n+1が割り付けられている。

【0077】同様に、オーディオデータ28の各オーディオユニットは、192aと192bとして表されるアクセスユニットカウント192を有している。ユニット

$m+1$ に対応するアクセスユニットカウント192aに割り付けられている値は、 $m+1$ であり、ユニット $m+2$ に対応するアクセスユニットカウント192bに割り付けられている値は、 $m+2$ である。アクセスユニットカウントの割り付けのその他の調整が本発明から離脱しない範囲で使用可能である。

【0078】図4において、図3のデータストリームのデコーダが示されている。このデコーダにおいては、マルチプレクスビットストリーム20がデジタルシグナルプロセッサ(DSP)98の制御の下に動作するデマルチプレクサ50に入力され、ビデオデータなどを分離し、ビデオデータをビデオバッファ100に供給する。その他の分離されたオーディオデータは、オーディオバッファ102に供給され、タイムスタンプは、ビデオタイムスタンプバッファ104に供給され、オーディオタイムスタンプは、オーディオタイムスタンプバッファ108に供給される。

【0079】図4の実施例において(図1の実施例でも同様であるが)、タイムスタンプとして全てのタイムスタンプを使用しようとすると、比較的大きく、かつ高価なビデオタイムスタンプバッファ104が必要となる。そこで、一度に1つのタイムスタンプのみを保持することができる、小さなタイムスタンプバッファを使用できるようにすることが好ましい。

【0080】タイムスタンプバッファを小さくすると、そこにデータが一杯記憶されているときは、その後入力されるタイムスタンプは破棄される。これは、タイムスタンプバッファに対する読み書きのポインタを必要としないため、バッファの管理が単純になる利点を有する反面、PLL制御が幾分遅くなる欠点を有している。

【0081】しかしながら、タイムスタンプは小さなエンコーダクロックエラーを再現するためにのみ使用されるのである。従って、(タイムスタンプに定義されている)デコーディングタイムからの付加的な偏差は、非常に小さな付加的なAVシンクエラーと、非常に小さな付加的なバッファエラーを発生させるに過ぎない。

【0082】例えば、従来のクリスタルがデコーダクロックを駆動するのに使用されるとき、数秒のクロック制御動作の遅延でも、90KHzクロックのせいぜい数サイクルの付加的なAVシンクエラーが発生されるに過ぎない。このように、これらのエラーは、半分のピクチャ周期シンクエラー(30Hzの場合、90KHzクロックの1500サイクル分)よりはるかに小さいので、タイムスタンプを関係するアクセスユニットに関連付けることは容易である。従って、PLL制御が幾分遅くなったとしても、実用的には、それ程、問題にはならない。

【0083】図4の実施例においては、図1のシステムにおける場合のように、初期化のため、最初のパケットヘッダのSCRがクロック118(MPEG1の場合、90KHz)により駆動される絶対時刻訂正レジスタ1

14に供給される。この絶対時刻は、ビデオPLL120とオーディオPLL122を駆動するために用いられる。ビデオPLL120とオーディオPLL122は、それぞれビデオバッファ100とオーディオバッファ102から、ビデオデータとオーディオデータを受け取るビデオデコーダ228とオーディオデコーダ230の時間設定を行う。

【0084】ビデオデコーダ228は、ビデオビットストリームから、(スタートアップ後)最初に読み出した(最初に遭遇した)アクセスユニットカウントを抽出し、それをローカルアクセスユニットカウントとしてレジスタ238に保持するビデオDSP234を有する。同様に、オーディオデコーダ230は、オーディオビットストリームから最初に読み出したアクセスユニットカウントを抽出し、それをローカルアクセスユニットカウントとしてレジスタ244に保持するオーディオDSP240を有している。

【0085】この実施例の動作は、図5のタイミング図を参照して、より理解することができる。時刻T1において、ビデオデコーダ228がフレームiのデコードを開始する。ビデオDSP234は、値i(最初のフレームヘッダからのアクセスユニットカウントを表す)によりレジスタ238を初期化する。時刻T1から1フレーム後の時刻T2において、DSP234は、次のフレームi+1をデコードし、レジスタ238をインクリメントし(i+1とし)、それを現在のアクセスユニットカウント(即ち、現在処理されているフレーム(いまの場合、フレームi+1)からのアクセスユニットカウント)と比較する。DSP234は、そのアクセスユニットカウントがレジスタ238に保持されている値と同一であることを確認し、これによりシステムが同期化する。

【0086】時刻T2からさらに1フレーム後の時刻T3において、DSP234は、フレームi+2をデコードするとともに、レジスタ238をインクリメントし(i+2とし)、その値を現在のアクセスユニットカウント(i+2)と比較し、その結果が適切であることを確認する。

【0087】時刻T4(DTS_{n+1})において、DSP234は、レジスタ238をインクリメントし、その値を(i+3)とする。しかしながら、フレームi+3のアクセスユニットカウント(i+3)が破損しているため、次のフレームi+4のアクセスユニットカウント(i+4)が検出される。その結果、レジスタ238の値(i+3)がアクセスユニットカウント(i+4)と比較され、これにより、両者が一致しないので、エラーが発生したことが検出される。

【0088】このとき、フレームi+4のデコーディングは、延期され、現在表示されているフレーム(i+2)が再度表示される。その後、時刻T5において、D

SP234は、フレーム $i+4$ のデコーディングを再開し、AVシンクエラーを訂正する。

【0089】図5に示すビットストリームが伝送エラーに起因して損なわれていなければ、MPEGスタンダードの3つの全てのパート（システム、オーディオおよびビデオ）を満足する。しかしながら、実際には、フレームヘッダが破壊されているので、ビデオのパーツとマルチプレクスビットストリーム（システム）のパーツを満足するものではない。また、受信されたビデオビットストリームは、フレーム $i+2$ に続いて、デコード不能のデータを有するため、かつ、フレーム $i+3$ が失われているため、これらのパーツを満足するものではない。さらに、この実施例の（図5の）マルチプレクスビットストリームは、フレーム $i+3$ のビデオパッケージが、タイムスタンプを有するが、検出可能なフレームヘッダを有しないため、パーツの条件を満足するものではない。これらのビットストリームは、エンコード時には、合法的なビットストリームであったものである。

【0090】この図3乃至図5に示す実施例は、AVシンクエラーがいかに検出され、訂正されるかを表すものである。この実施例においては、消失ピクチャエラーは、1ピクチャ周期後に直ちに訂正される。しかしながら、よりインテリジェントなビデオデコーダは、（例えば）静止画（または殆ど動かない）ピクチャシーケンスが発生するまで、訂正手続を延期することができる。同様のことは、オーディオの場合にも可能である。オーディオアクセスユニットをスキップしたり、コピーすることは、オーディオトラックの無音部分の区間において行うのが好ましい。

【0091】アクセスユニットの消失または付加に拘らず再生を継続するためには、バッファの容量は（エラーが少ないシステムにおける場合より）、より大きくなされるべきである。これにより、アンダフローを抑制し、かつ（アプリケーションによっては）、オーバフローを抑制することができる。アクセスユニットの消失の問題は、例えば、高い伝送レートエラーと極めて低いビデオビットエラーを有するアプリケーションにおいて最も発生しやすい。この場合、そのエラーがフレームヘッダで発生する可能性が高くなる。

【0092】望むならば、あるいは必要とするならば、編集されたビットストリームの再生を可能にするために、“不連続ビット”をアクセスユニットカウントフィールドに付加することができる。図6の実施例は、ビットストリームをいかに編集するかを表している。この場合、編集者は、1フレームを除去することを決定する。これにより、フレームは、連続するアクセスユニットカウントを有していない。

【0093】即ち、図6において、AACはアクセスユニットカウントを表し、編集前の状態においては、1ずつインクリメントされている。しかしながら、フレーム

（ $i+1$ ）を除去するように編集すると、AACは i から $i+2$ に変化し、不連続となる。そこで、フレーム i 以前のヘッダや、フレーム $i+3$ 以降のヘッダにおいては、不連続ビットDBを0にするが、フレーム（ $i+2$ ）のヘッダにおいては、不連続ビットDBを1とし、不連続であることを表す。

【0094】デコーディングシステムは、不連続ビットDBが0の場合、図5に示したように動作するが、不連続ビットDBの1が検出される毎に、ローカルアクセスユニットカウントを再度初期化する。これにより、編集されたビットストリームが誤って処理されることが防止される。

【0095】但し、アクセスユニットの消失または付加が、そのビットストリームの編集ポイントにおいて発生した場合、不連続ビットを用いてフレームの不連続を検出することができない（換言すれば、不連続ビットDB=1は、1アクセスユニット周期の間、デコーディングシステムのアクセスユニットカウントチェック機構を不能にする）。

【0096】本発明は、多くの異なる形の実施例を受け入れることができるが、図示および詳細な記述は、特定の実施例についてのみ行われる。この開示は、本発明の原理の例であり、本発明を、示され、かつ記述された特定の実施例に限定するものではない。例えば、本発明は、CD-ROMおよびインタラクティブCDアプリケーションに用いることが可能である。また、実施例は、MPEGのマルチプレクスシステムのためのものであるが、本発明は同様のシステムに適用することが可能である。

【0097】このように、本発明の目的、意図、利点を完全に満足する装置が上述されたことは明かである。本発明は、特定の実施例に関連して記述されたが、上記した記述に照らし、多くの代案、修正、置換および変更が当業者にとって自明である。従って、本発明はこのような全ての代案、修正および変更が添付した特許請求の範囲内に含まれるものである。

【0098】

【発明の効果】以上の如く本発明によれば、アクセスユニットヘッダの消失または過剰を検出することができる。また、アンロックシステムにおいて、同期化を維持することができる。さらに、編集されたマルチプレクスビットストリーム、および編集されたエレメンタリビットストリームを再生することができる。

【図面の簡単な説明】

【図1】関連する発明のアンロックデコーディングシステムの図である。

【図2】図1のアンロックデコーディングシステムの動作のタイミング図である。

【図3】本発明のアクセスユニットカウントを組み込んだデータストリームの図である。

【図4】本発明のデコーディングシステムのブロック図である。

【図5】図4の実施例の動作を説明するタイミング図である。

【図6】ビットストリームの編集を表す図である。

【図7】MPEGデータストリームの図である。

【図8】シンタックスの変更を説明する図である。

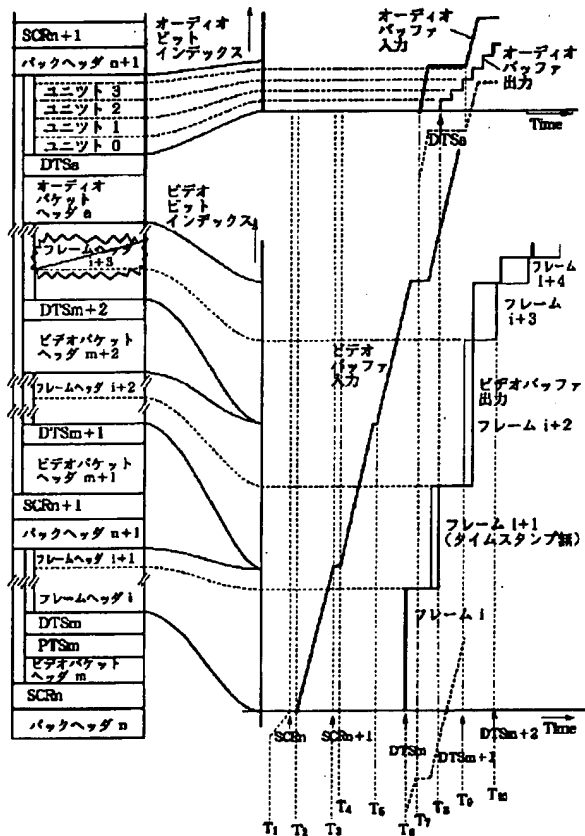
【図9】図8のデータストリームを生成するシステムのブロック図である。

【符号の説明】

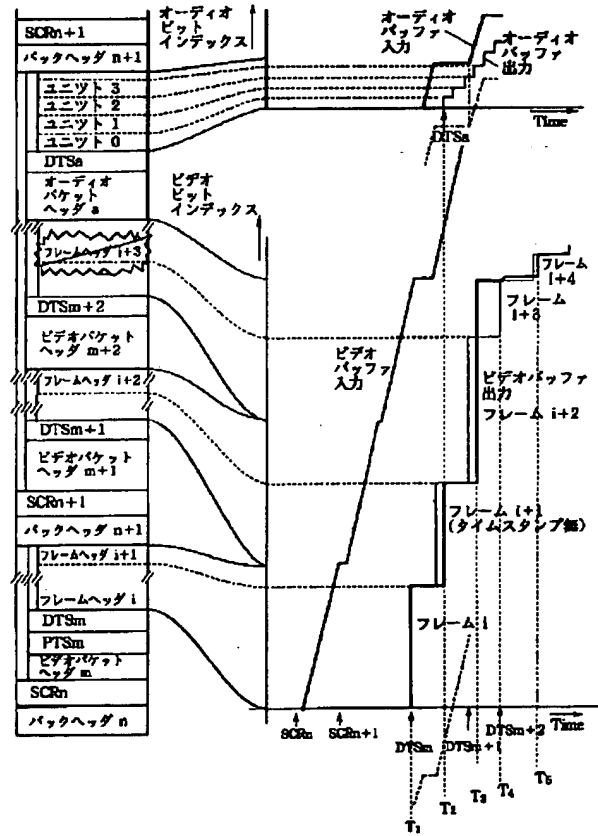
20 マルチプレクスビットストリーム
36 ビデオビットストリーム
50 デマルチプレクサ
98 デジタルシグナルプロセッサ
100 ビデオバッファ
102 オーディオバッファ

104 ビデオタイムスタンプバッファ
108 オーディオタイムスタンプバッファ
114 絶対時刻訂正レジスタ
118 クロック
120, 122 PLL
128 ビデオデコーダ
130 オーディオデコーダ
190, 192, 192a, 192b アクセスユニット
カウンタ
228 ビデオデコーダ
230 オーディオデコーダ
234 ビデオDSP
238 レジスタ
240 オーディオDSP
244 レジスタ

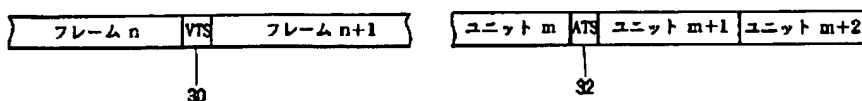
【図2】



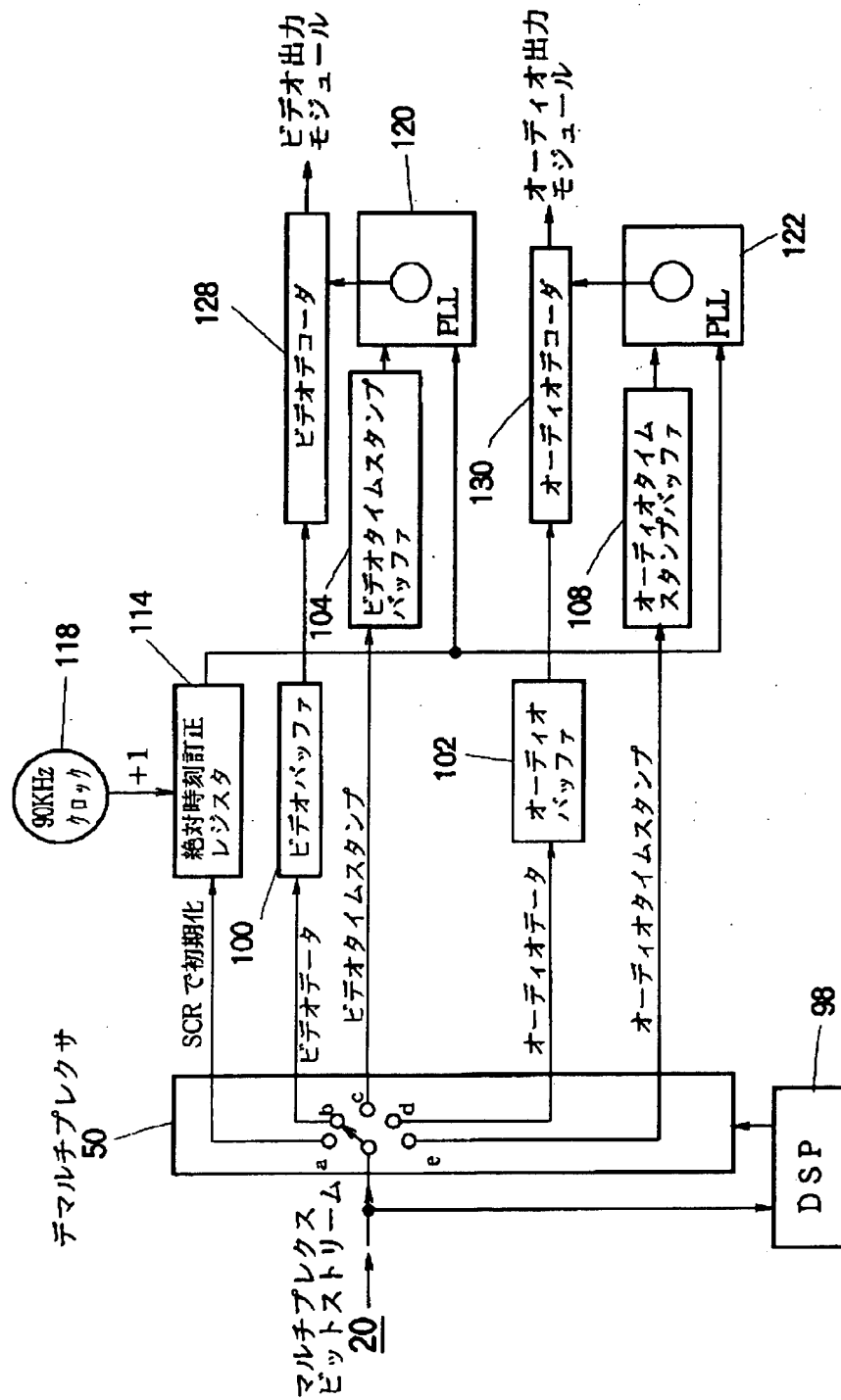
【図5】



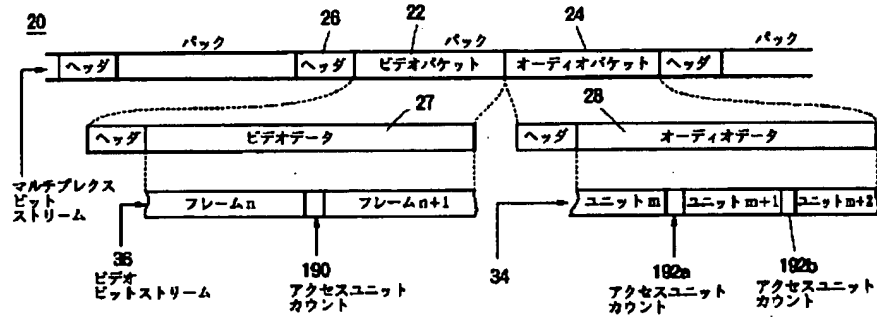
【図8】



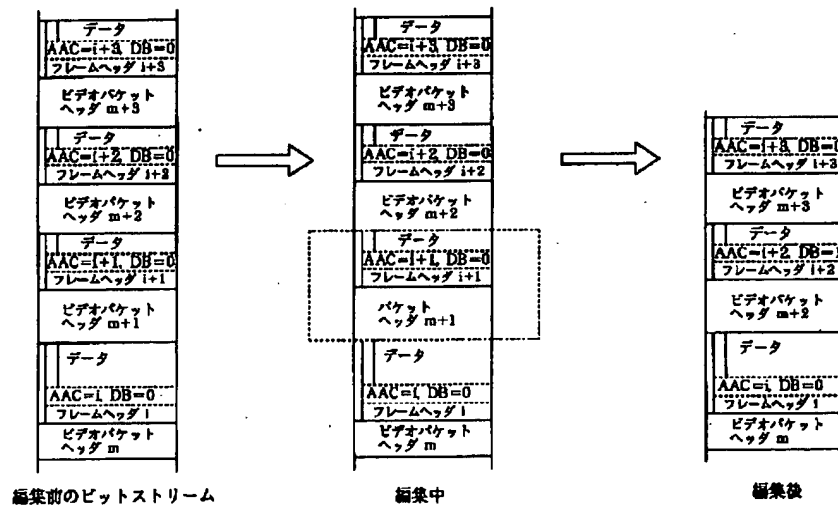
【図1】



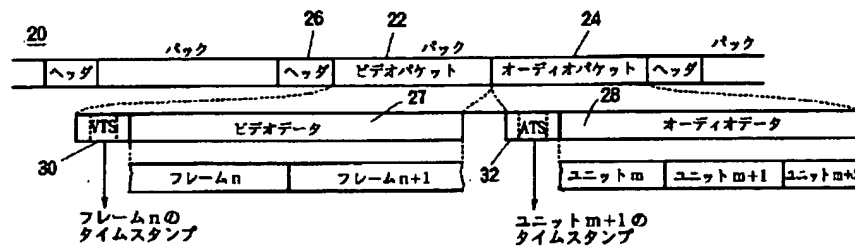
【図3】



【図6】



【図7】



【図9】

